A Method to Help Minimize the Cost of Quality

Lissa Galbraith

Mail Stop 125-161, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109-8099

INTRODUCTION

Part of the Jet. Propulsion Laboratory (JPL) mission is to ensure that our programs are affordable and of high value to our customers. A component of the NASA vision is to develop technology to keep America capable and competitive. This project directly addresses these issues. In principle, one should be able to determine a direct quantitative relationship between: specific defects, associated failure mechanisms, and Quality Assurance (QA) requirements for inspection and testing. In practice, it is very difficult to determine these relationships. The identification of such relationships and the correlation of QA requirements and value added (cost of quality) to the product was determined. A key project objective was to minimize the number of quality indicator variables necessary to monitor and control the electronics assembly process.

We describe a general methodology to design for producibility and reliability for very small volume production runs. In cases where the entire volume for fabrication is less than five products, traditional Statistical Process Control (SPC) is inadequate due to reliance on statistics of much larger volumes and the Central Limit Theorem. Data acquisition for process parameter estimation from such a small sample size is difficult; however, it is critical to producing a high reliability product.

Small volume fabrication is often as expensive or more expensive than high volume production to achieve acceptable performance levels. Cost factors such as material, assembly time, and safety remain important- parameters for small volumes. Manufacture of circuit card assemblies and system units at JPL is such an example. The need for very high functionality, safety, and reliability drives design and fabrication costs up, while the small total volume and individual component expense leave little latitude for error.

TAGUCHI METHODS

Traditional approaches fail to address the needs of small volume high reliability electronics manufacturing for several reasons :

1. The traditional methods invoke the Central Limit Theorem, implying an assumption of a normal distribution with greater than 30 data points.

2. The traditional methods assume that repair is a reasonable possibility, even after the unit is in the field. There is an extremely high price for NASA to retrieve a satellite when it. fails. Medical electronics also cannot depend on going in after the failed component.

3. Traditional methods do not adequately address high reliability requirements which stress safety.

4. Traditional methods do not accommodate the very long expected lifetimes of NASA products.

This method is based on the Taguchi Loss Function [1, 2].The Taguchi loss function involves a different philosophical approach to quality: the further the product features are from the target value, the greater the defined loss. Traditionally, in the US, when products are within tolerance specifications, products are passed, then shipped. Genichi Taguchi defines loss as functional variation plus cost caused by the product being defective. The Taguchi Loss Function is defined as the mean square deviation of specific features of a product from the target values of these features or:

$$L(y) = k(y - m)^2 \qquad (1)$$

where y . specified feature characteristic
m = target value
k = proportionality constant,

$$k = \frac{\text{cost of a defective product}}{(\text{tolerance})^2} = \frac{A}{\Delta^2}$$

As the deviation from the target increases, an increase in loss of performance is seen. This cost may be a decrease in expected product lifespan or a decrease in the expected Mean Time Between Failures (MTBF). The Taguchi loss function remains valid with very small. sample sizes. The mean square deviation of a specific feature from its target value may be used to estimate the mean performance loss of Equation (1), where the Mean Square Error (**MSE**) or mean square deviation from the target value is defined as:

$$MSE = \text{mean value of } (y - m)^2$$

The Taguchi loss function may then be simplified to: L = **k(MSE)**.

Economic safety factors: account for the cost implications of variations in the product feature of interest. The economic safety factor = ϕ, where

ϕ = [(mean cost when specific product feature exceeds product functional limits) ÷
(mean cost when same product feature exceeds design tolerance specification)]$^{1/2}$

The numerator is designated to be A_0 and the denominator to be A. The economic safety factor is then:

$$\phi = (A_0/A)^{1/2}$$

If the defective part is reworked during assembly, then A = cost of rework or scrapping the product.

Taguchi advocates putting more time up front., in the design of the product, while also trying to continually improve the assembly process itself. His recommendation is to maximize the signal to noise ratio (S/N) to improve processes. Signal factors are the intended inputs to the process. Noise factors are uncontrollable error factors. The process is said to be "functionally robust" if the design intent is satisfied for a wide range of part features. Rather than attempting to eliminate or minimize noise factors, the design can be adapted to be less sensitive to these factors.

PRINTED WIRING ASSEMBLY EXAMPLE

PWA signal factors include voltage, current, component. dimensions, solder viscosity at a given time, vapor phase sump temperature, etc. Ranges of the signal factors to test for process improvement may be selected from the chosen design levels. PWA noise factors include dirt, solder voids and bridges, chip movement during reflow, humidity, etc. Noise ranges may be ascertained by observation.

ANALYSIS OF EXISTING DATA

The table below summarizes the basic data gleaned from a few existing laboratory qualification boards (five total) and a board being assembled now for experiments. The data refers to individual designs rather than boards. NA = not available. NYA = not yet available. Placement misalignment refers to the number of devices cited for misalignment after reflow causing the device to be closer to its nearest neighbor than the PWA overall designed minimum spacing between devices. Misalignment is counted for any number of leads overhanging a solder pad.

	Board 1	Boards 2 -- 6
Total Number of Devices Of Devices	84	99
Area (in^2)	41.65	54.44
Mean Device Density ($parts/in^2$)	2.02	1.79
Minimum Spacing Between Devices	1.5	1.5
Max i mum Distance to Neutral Point	37.5	3
Minimum Lead Pitch	20	20
Placement Misalignment.	NYA	10

Process procedures are in control and well documented. At this time, it appears that the design for assembly policies encourage optimizing the process so that the process will be insensitive to design flaws. While this should be the general policy for continuous process improvement, a more cost-effective approach would be to attempt to optimize the design for assembly and for insensitivity to process flaws. We typically have much more control over design than over processes. Some process challenges could be significantly reduced or eliminated by encouraging better communication (as part of concurrent engineering) during the PWA design phase.

DESIGN OF EXPERIMENT

The example experiment design is a partial factorial design, as described below.

Basic Design: 2^k = n = number of runs, k = number of factors,
2 . number of levels, 2^3 : 8 runs
+ . high level and - . low level for a factor

Determine whether these factors do indeed influence the signal as hypothesized. Run an Analysis of Variance (ANOVA) first to determine the significance of these factors at their chosen levels. Check ANOVA assumptions for validity in your assembly situation. Major ANOVA assumptions are:

1. Process is in control
2. Population distribution is normal
3. Errors are homogeneous

Assumption 1 can reasonably be said to be true. Assumption 2 can be said to be true if the substitution of the t-distribution is made for the normal distribution, to account for the low volume of samples. Assumption 3 is made initially and will be rechecked as ANOVA residual and the Sum of Squares (SS) values are made available.

ultra low volume production data acquisition was studied to maximize the information to be gained from the data and minimize the total volume required and cost of acquisition. In order of preference these methods are:

1. Examine existing historical data
2. Re-analyze and possibly partially reprocess rejected product
3. Run and analyze test coupons
4. Run and analyze product produced for these experiments.

The signal and noise factors for the example were:

Signal (Experiment Output or Response)	Influencing Factor
* Solderability	* Tinning
* Solder fillet formation	* Lead forming
* Coplanarity	* Lead forming
* Solder Joint Failure	* Thermal mismatch, assuming board design OK
* Cleanliness	* A = Minimum spacing between devices
	* B = Minimum device standoff from board
	* C . Maximum distance to neutral point
	* D = Minimum lead pitch

Noise

* Chip movement during reflow
* Minor flux residue

using the cleaning signal as an example with the influencing factors as defined above, an example DOE is:

Run Number	A	B	C	D
1				
2	+			+
3		+		+
4	+	+		
5			+	+
6	+		+	
7		+	+	
8	+	+	+	+

This DOE is a partial factorial design with confounded interactions. It was determined that interactions among influencing factors could be considered negligible or of questionable interpretation. The S/N ratios, η, were computed from the experiment. results.

$$\eta = 10 \log(1/r) \; [(S_\beta - V_e)/V_N]$$

where r . magnitude of input signals,

S_β = sum of squares for each signal factor

V_e . mean square error of nonlinearity where

$V_e = S_e/(2k - 2)$,

2k - 2 = number of degrees of freedom

k . proportionality constant defined above

S_e = sum of squares of the error term $= SS_{error} = S_T - S_\beta - S_{N*\beta}$

V_N = error term of nonlinearity and linearity

Having computed the S/N ratios, the next step is to improve the process. First, estimate the proportionality constant between component and PWB parameters. Tune controllable process factors to increase the S/N ratio. Process tuning may involve improving factors such as cleanliness, ESD, component values, solder paste curing time, etc. The next step is to adjust design features to be less sensitive to noise factors and closer to target values, for example, choose a different type of IC (any component) which has a better seal or dissipates heat more successfully. To summarize the five steps:

1. Identify signal and noise factors and their ranges.
2. Using fractional replication in a design of experiment, assign the design signal factors to experiments.
3. Compute S/N rat.ios from the experiment results.
4. Improve process conditions and estimate the proportionality constant between component and PWB assembly parameters.
5. Adjust design features to be less sensitive to noise factors and closer to target values.

There is a need for high reliability PWB assembly processes for aerospace, military and medical applications. This approach to improving design for producibility and reliability of circuit and electronic system assembly processes can be utilized as a beginning framework.

Academic, military, and commercial electronics assembly experts are part of an evaluation team which has assessed project progress during and after completion of the research. There is no doubt that a new approach to high reliability quality assurance is mandatory if JPL design for and production of electronics assemblies is to become more cost-effective. The technological advances developed from this research will significantly contribute to this effort .

BIBLIOGRAPHY

1. Barker, Thomas B., *Engineering Quality by Design, Interpreting the Taguchi Approach,* ASQC Press, New York, 1990

2. Taguchi, Genichi, *Taguchi on Robust Technology Development*, ASME Press, New York, 1993

such reasons as software development, integration, or operations the answer may well be, "yes."

Consider again the example of C, UNIX, and RISC. If the lowest common denominator were not C but IBM 370 assembly code, we could never have had the RISC revolution. There would be a 370-on-a-chip but the great strides in performance could not have been realized due to that chip's complexity. Higher level standards (C and UNIX) allowed a sufficiently large trade space that RISC could be "discovered".

Recommended Practices

Reward systems which encourage individuals to focus on only one program are self-defeating in the long-term. Instead, a "standards culture" must be nurtured. For each interface, we must ask "What standards are applicable?" The industry or, at the very least, each institution should select a small family of interface standards which satisfy a variety of needs and then use them. Emphasis should be placed first on international and national standards followed by Federal government and NASA standards. The existence of institutional standards and project "standards" should always be questioned, In fact, while a project may select a particular component for ease of procurement, a "project-specific interface standard" is an oxymoron. It should be emphasized that the greatest benefits of standards is in their contribution to a lasting infrastructure; standards which are specified for what is effectively a point-design fail to contribute in the long-term.

Part of the standards culture must be the recognition that "good enough" is perfectly valid engineering practice. In fact, in the commercial world, this is exactly what is desired. If an existing standard is "almost right" for a particular job, then one must ask whether it is "good enough" (perhaps by slightly relaxing some requirement). "Perfection" is the enemy of "good enough". The former can never be achieved; the latter is often readily achieved and implemented.

If no existing standard will suffice, there is probably a compelling reason why a broad class of applications has similar requirements. In this case, a new standard should be contemplated and perhaps pursued. If an existing standard can serve as the basis for the new standard, so much the better. However, it must be recognized that a modified standard is no longer a standard and most if not all the benefits associated with a standard are lost (e.g., the phrase "lew-power 1553 bus" should only be used to refer to a "low-power implementation of a 1553-conforming bus"). In the long run, this cultural base benefits both the customer and the vendors.

Standards in Use

At present, there are relatively few standards that enjoy acceptance by the space community, in large part due to the attitude that each space mission requires an optimal solution for each interface. Possibly the only moderately high-level interface standards in wide use on spacecraft are the MIL-STD- 1553B and MIL-STD-1773 serial cable busses. Additionally, VMEbus is gaining some acceptance as a low-cost, parallel backplane bus. The RS-232, RS-422, and RS-485 standards are often used for signaling but, unfortunately, many members of the community cite their use as if they are standards on par with 1553. In reality, they are only the electrical specification for a physical layer and saying "we're using RS-422° is similar to saying "we're using five-volt logic levels."

A number of programs which could have resulted in standards are often cited as though they are standards when, in fact, they are not. Among these are GVSC, ASCM (or "ASCM bus"), and RH-32 (there are two, incompatible RH-32s).

The table in the appendix summarizes a number of existing and emerging formal, open interface standards which may be applicable to space missions.

Creating New Standards

Standardization requires that there be a clearly identified marketplace need, a plethora of options (probably many previous designs), a well-understood option space (a maturity of the designs), and a recognition that the options have more commonality than differences (a convergence of thought). Finally, there must be the ability to superset the option space to encompass most of the requirements of the various previous designs without creating an unwieldy solution (such as might lead to excessive cost).

What to Standardize

The most important things to standardize are those for which the potential cost-savings are greatest. A good place to start is with things that vendors see as being constantly reinvented or re-specified such as:

System Architectures (e.g., OS1)
Busses (Backplane, Serial, Analog)
Sensor and Instrument interfaces
Control Languages
Software Environment
Domain-Specific Software
Software Reusability
Telemetry Standards
Physical Environment
Mechanical Form-Factors
Fault Tolerance
Operating System Services (e.g., POSIX)
programming Languages (Ada, C)
Applications (e.g., attitude control)
Communications Protocols (e.g., CCSDS, TCP/IP)
Testability Design

The following requirements must be considered for all standards which arc to be proposed for space data systems:

- General spacecraft functional requirements such as space environment, low-power, low-mass, high performance, fault tolerance, interoperability, reliability.

- Project development requirements such as software and hardware heritage, reuse, and testability.

- Market acceptance requirements such as modularity, flexibility, generality, evolution.

Note that in the previously listed areas for potential interface standards, there are a number of items which might not appear to be interfaces at all. Is domain-specific application software really a candidate for standardization? If wc instead ask, "is it a candidate for interface standardization?", the answer is an emphatic "yes !" There is no inherent reason why spacecraft applications cannot enjoy the same degree of interface uniformity as enjoyed by data bus standards.

How to Standardize

When it has been identified that a standardization effort would be beneficial, the starting point for the standard's definition should be the lessons learned (hopefully in a documented form) from previous efforts. Generally, these indicate both the pitfalls to avoid and the feature required to provide wider applicability than the prior point-designs. If it appears that there is much commonality across applications but that no single standard would be justified for all of them, then the standard should define a framework within which profiles are defined for specific applications. For instance, allowing slight distinctions between the requirements which are appropriate for a planetary probe and a slightly different set for a launch vehicle. Because standards take time to define, consideration must be given to allowing technology evolution (so that a standard is not obsolete before it is accepted) particularly by having the ability to extend the standard while retaining backward compatibility with the initial standard. All this must be done while minimizing the number of options supported in order to reduce complexity, to reduce proliferation of variants, and to increase interoperability (lest interoperability be limited to a small intersection of the feature sets).

The best way to get a standard through the approval process is to go to the standards body with as complete a draft as possible. If the potential user (or users) has already done a lot of industry research and standard definition, there is less likelihood of the committee-consensus process taking an inordinate amount of time. The draft document should also include an integral compliance test and evaluation suite. In a very real sense, this test suite will become the actual standard while the formal (officially recognized) standard will simply be the specification of the test.

Finally, if a standard involves the use of a proprietary innovation, its owner must be convinced that relinquishing proprietary rights to it serves long-term interests better. This generally requires that seeing that the market will be larger as a result of standardization and that market shares will be relatively stable for some period of time after the standard is released.

Once a standard is approved, it must be accepted by the marketplace based on its own merits; the use of a particular standard cannot be mandated without the possibility of incurring unexpected costs that outweigh the expected savings. Good standards are used voluntarily; bad standards are only used coercively. For example, contrast the widespread use of C in virtually every field with the use of Ada in government contracts and little else. There are no bad standards in a free-market, commercial world; bad standards die. As a corollary, if standards were not good things, they would all cease to exist.

What do good standards have in common? First and foremost, they recognize the needs of the customer. The only way to ensure that real needs are met is to involve potential customers in the standardization process, preferably very early. Typically, vendors instigate standardization efforts but they must actively solicit input from their expected customers.

Good standards are not all things to all people. The inclusion of outlying applications is bound to increase complexity and thus cost. Good standards recognize that incremental improvement is necessary to stay competitive; they do not stifle innovation. They allow some degrees of freedom so that there is incentive for vendors to introduce competitive products, to add value to the marketplace.

Ada is disliked from a technical standpoint because the working group wrongly identified the government as Ada's customer where, in actuality, software engineers and programmers are the customer, Because the committee tried to solve a very large set of problems, Ada is overly complex and its compilers are consequently large and slow.

Summary Conclusions

Suppliers want the systems houses to select standards and then use them repeatedly. There is a big enough market among the spacecraft and launch vehicle manufacturers to support standard product lines but this can only occur if potential customers can agree on a small number of interface standards. A wide consensus reached by a customer-sponsored forum (e.g., SATWG, the NASA-sponsored Strategic Avionics Technology Working Group) would foster aerospace industry standard use. Advanced technology insertion efforts are a prime candidate for standards development,

We live in a new era -- one in which full-custom designs will be increasingly difficult to justify. Without wholeheartedly pursuing the definition and use of standards, the industry will not be able to compete for scarce funds and will cease to exist as we know it.

Acknowledgements

This paper is based on observations made during various standardization activities and related work performed by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Thanks to S. Chau, J. Brutocao, G. Man, and D. Rennels for their thoughtful comments.

Appendix

The following table summarizes some standards which are in use in spacecraft avionic systems or which may be of interest to those contemplating using non-traditional ones or creating new standards. This list is by no means complete. The inclusion of some standards (PC/1 04) and exclusion of others (e.g., ISA bus) reflects the biases of this author as to what is "interesting" in the context of spacecraft applications. And even that opinion is subject to change...

Standard Designation	**Name** Date of Most Recent **Revision**	General Classification **Short Description**
Standards in Spaceborne Applications		
MIL-STD-1553B	**Digital** Time Division Command/Response Multiplex Data Bus, Notice 2 8 September 1986.	Serial Command- Response Databus (ISO Physical, Link layers). Widely used in spacecraft. Required transformer isolation results in relatively high power consumption.
MIL-STD-1773	Fiber Optics Mechanization of an Aircraft internal Time Division Command/Response Multiplex Data Bus, Notice 1 2 October 1989.	Serial Command-Response Databus (ISO Physical, Link Layers). Fiber-Optic Physical Layer mechanization of MIL-STD-1553B.
ANSI/IEEE Std 1014-1987	A Versatile Backplane Bus: VMEbus 11 September 1987	Parallel Backplane Bus (ISO Physical, Link Layers). Widespread commercial use and military ruggedization. Synchronous 32-bit data transfers at 16 MHz yield 48 MB/s throughput.
EIA RS-232D		Electrical Interface (Part of ISO Physical Layer). Unbalanced line driver specification; uses wide voltage swings and dead-band around OV to achieve noise immunity. Very widespread use,
EIA RS-422A	Electrical Characteristics of Balanced Voltage Digital Interface Circuits December 1978	Electrical Interface (Part of ISO Physical Layer). Balanced (differential) line driver specification commonly used for point-to-point communication in noisy environments.
EIA RS-485	Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems April 1983	Electrical Interface (Part of ISO Physical Layer). Balanced (differential) line driver specification which allows multi-drop (bus) configurations. Protections specified to ensure non-destructive multi-transmitter conflicts.
Aircraft Data Communications Standards		
ARINC Specification 429-14	Mark 33 Digital Information Transfer System (DITS) Aeronautical Radio, Inc. 10 March 1993.	Serial Data Interchange. Older aircraft data exchange largely superseded by ARINC 629. Single-transmitter multiple-receiver architecture. Standardizes across the industry word formats for common data (e. g., inertial reference),
ARINC Specification 629-2	Multi-Transmitter Data Bus Aeronautical Radio, Inc. 16 October 1991.	Serial Cable Bus. Intended to reduce cabling complexity of 429 systems. Line couplers allow complete maintainability for aircraft but are too heavy anti power-consumptive for spacecraft.

ARINC Specification 659	Backplane Data Bus for integrated Modular Avionics (Draft) Aeronautical Radio, inc. May 1991	Parallel Backplane Bus. Moderate-throughput bus for intra-cabinet communication.
SAE AS4074.1	Linear Token Passing Multiplex Data Bus (HSDB), September 1988	Serial Cable Bus/LAN (1S0 Physical, Link Layers). Optical bus at 50 Mb/s. Implements logical ring. Links Pi-Bus chassis in F-22.
SAE AS4075	High-Speed Ring Bus (HSRB) April 1992	Serial Cable Bus/LAN (1S0 Physical, Link Layers). Fault-tolerant, token-passing network with consideration of military needs.
SAE AS4710	Pi-Bus	Parallel Backplane Bus. Air Force sponsored. Fault-tolerance emphasized. Tends toward high power.

Existing Computing Standards

ANSI X3.131-1986	Small Computer System Interface **(SCSI)** 23 June 1986	Parallel Cable I/O Bus. Moderate-throughput (1 MB/s), 2.-drop 8-bit bus intended for microcomputer peripherals of earl y- 1980's vintage. Limitations addressed by SCSI-2.
ANSI X3.139-1987	Fiber Distributed Data Interface (FDDI) - Token Ring Media Access Control (MAC) 5 November 1986	Serial Cable Bus/LAN (1S0 Physical, Link, Network, Transport Layers).
ANSI X3.148-1988	Fiber Distributed Data Interface (FDDI) - Token Ring Physical Layer Protocol (PHY) 30 June 1988	Serial Cable Bus/LAN (1S0 Physical, Link, Network, Transport Layers).
ANSI/IEEE Std 488.1-1987	Standard Digital Interface for Programmable Instrumentation 2 February 1988	Parallel Cable 1/0 Bus (1S0 Physical, Link, Network, Transport Layers). Used primarily for instrumentation; also known as GPIB -- General Purpose instrumentation Bus.
IEEE 754-1987	Standard for Binary Floating Point Arithmetic	Defines an interface between user software and floating point hardware or software.
Pc/104	Standard for Embedded-PC Modules	Defines a small, rugged form-factor for the venerable PC/AT (ISA) backplane bus. Useful for small embedded applications.

Standardization Efforts

IEEE P896. 10	Futurebus+, Space Profile	Parallel Backplane Bus (1S0 Physical and Link Layers). High-throughput (100 MB/s) backplane bus with specializations for space environment and missions, e.g., thermal and fault-tolerance.

IEEE P1101.7	Mechanical Standard for Space Applications Module, Extended Height Format E Form Factor	Mechanical. Specifics a conduction-cooled electronic module of 150 mm x 210 mm with a 300 or 396 pin connector on the 150 mm side.
IEEE PI 156.4	Environmental Specification for Spaceborne Electronic Modules	Environmental. Specifics three levels of space environments and the test methods for qualification and acceptance of electronics modules which are to operate in them.
IEEE P1394	High-Speed Serial Data Bus	Serial Cable and Backplane Bus (1S0 Physical, Link, Network, Transport Layers), Inexpensive (commercial) short-haul chassis-chassis connection at 100, 200, or 400 Mb/s. Six wire cable. Backplane version used by Futurebus+ alternate serial bus.
IEEE FODB	Spaceborne Fiber-Optic Data Bus	Serial Cable Bus (1S0 Physical, Link, Network, Transport Layers). 100 Mb/s payload data bus. Existing FODB contracts will be used as basis,
SAE AS4848	Digital Time Division Command/Response Multiplex Data Bus	Serial Conlmand-Response Databus. SAE adoption and augmentation of MIL-STD-1553B and STANAG 3838. Intended additions: dual-rate, 4 Mb/s, extended addressing, block transfers, low-power transceivers.
SAE AS4893	Generic Open Avionics Architecture	Architecture. Derived from a JSC-sponsored Lockheed contract (Space Generic Open Avionics Architecture). Attempts to codify good avionics system practice. Recommends system structure, terminology and standards.
SAE J1939	Heavy Vehicle Communications Interconnect	Serial Cable Bus (1S0 Physical, Link Layers). Vehicle control for heavy vehicles, e.g., tractor-trailers. Uses Controller Area Network protocol, Useful as an example of an end-to-end application standard.
ANSI X3.131-199X	Small Computer System Interface (SCSI-2) 375R Revision 10K, 28 April 1993	Peer-to-Peer Parallel Cable 1/0 Bus. Fast (1O MHz transfer rate) bus supporting 8, 16, and 32 bit transfers (most systems at 16). Up to 16 devices per bus.
ANSI X3.131-199X	Small Computer System Interface (SCSI-3)	Peer-to-Peer Parallel Cable 1/0 Bus. High-speed, fiber-optics, and functional improvements to SCSI-2. Effort initiated mid-93,
AIAA GN&C		AIAA GN&C Committee on Standards is attempting to establish recommended practice for the use of standards in spacecraft GN&C.